UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/652,135 | 08/29/2003 | Philip E. May | CML00769D | 1155 |

33117 7590 06/12/2007
LEVEQUE INTELLECTUAL PROPERTY LAW, P.C.
221 EAST CHURCH ST.
FREDERICK, MD 21701

| EXAMINER | |
|---|---|
| MEONSKE, TONIA L | |
| ART UNIT | PAPER NUMBER |
| 2181 | |
| MAIL DATE | DELIVERY MODE |
| 06/12/2007 | PAPER |

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 10/652,135 | MAY ET AL. | |
| | Examiner | Art Unit | |
| | Tonia L. Meonske | 2181 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *02 May 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) *1-12* is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *13-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 August 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *See Continuation Sheet*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

Continuation of Attachment(s) 3). Information Disclosure Statement(s) (PTO/SB/08), Paper No(s)/Mail Date :2/16/07,6/1/06,5/5/06,6/21/04,8/29/03.

## DETAILED ACTION

### *Election/Restrictions*

*1.* Applicant's election without traverse of claims 13-24 in the reply filed on May 2, 2007 is acknowledged.

### *Specification*

2. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

### *Claim Rejections - 35 USC § 112*

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

4. Claims 13-24 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

5. Referring to claim 13, the limitation "a...loop having a loop body but no prolog instructions" is unclear. According to applicant's specification, paragraph [0003], instructions at the start of the loop are described as the prolog instructions. It is not understood how a loop can have no instructions at the start of the loop. Each loop must have instruction(s) at the start of.the loop otherwise the loop would contain no instructions and it wouldn't actually be a loop. It would be nothing. Appropriate correction is required. For the purposes of examination this limitation is interpreted as "the loop body instructions enter a pipeline with instruction(s) already executing in the pipeline stages".

6. Claims 14-19 are rejected for incorporating the defects of claim 13.

7. Referring to claim 19, the limitation "wherein the loop has not epilog instructions" is unclear. According to applicant's specification, paragraph [0003], instructions at the end of the loop are described as the epilog instructions. It is not understood how a loop can have no instructions at the end of the loop. Each loop must have instruction(s) at the end of the loop otherwise the loop would contain no instructions and it wouldn't actually be a loop. It would be nothing. Appropriate correction is required. Appropriate correction is required. For the purposes of examination this limitation is interpreted as "the loop body instructions exit a pipeline with instructions following the loop executing in the pipeline stages".

***8.*** Referring to claim 20, the limitation "a...loop having a loop body but no epilog instructions" is unclear. According to applicant's specification, paragraph [0003], instructions at the end of the loop are described as the epilog instructions. It is not understood how a loop can have no instructions at the end of the loop. Each loop must have instruction(s) at the end of the loop otherwise the loop would contain no instructions and it wouldn't actually be a loop. It would be nothing. Appropriate correction is required. Appropriate correction is required. For the purposes of examination this limitation is interpreted as "the loop body instructions exit a pipeline with instructions following the loop executing in the pipeline stages".

***9.*** Claims 21-24 are rejected for incorporating the defects of claim 20.

***Claim Rejections - 35 USC § 102***

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

11. Claims 13-24 are rejected under 35 U.S.C. 102(b) as being anticipated by Hennessy and Patterson, Computer Architecture A Quantative Approach, 1996, Morgan Kaufman Publishers, Inc., Second Edition, pages 239-247 (hereinafter referred to as Hennessy).

12. Referring to claim 13, Hennessy has taught a method for executing a pipelined program loop having a loop body but no prolog instructions on a processor (This limitation is interpreted as "the loop body instructions enter a pipeline with instruction(s) already executing in the pipeline stages" (see the 112 rejection above). page 240, The loop enters the pipeline with at least one instruction executing in the pipeline, A[1] = A[1] + B[1];), the processor comprising a plurality of functional units coupled through an interconnection switch and controlled by a controller (pages 246 and 247, at least the multiply unit, integer unit, add unit and divide unit are the functional units), each functional unit of the plurality of functional units having at least one input for receiving an input data value and an associated input data validity tag (pages 246 and 247, Fj, Fk are the source registers that are input to the functional units from functional units Qj and Qk. Rj and Rk is the validity of the input data.) the method comprising: executing the loop body for a plurality of iterations (page 240, iterations of the loop); and at each iteration of a plurality of iterations: determining if the input data values are valid by

checking the associated input data validity tags (page 246 and 247, Rj and Rk, The scoreboard is checked for each instruction to ensure all data values are valid before executing the instructions.).

***13.*** Referring to claim 14, Hennessy has taught a method in accordance with claim 13, as described above, and wherein a functional unit of the plurality of functional units includes a result register for storing an intermediate result (page 246 and 247, When Fj or Fk are dependent on a prior operation.) and an associated output data validity tag (page 246 and 247, Rj and Rk indicate the readiness of operands dependent on other operations executed and output from the various functional units.), the method further comprising: at each iteration of the plurality of iterations: if all of the input data values are valid, performing a functional operation on the input data values (page 246 and 247, When the flags of a functional unit indicate that all data values are ready, then the functional unit performs it operation on the input values.), storing the result of the functional operation in the result register and setting the associated output data validity tag to indicate that the intermediate result is valid (page 246 and 247, For example-F2 in the multiply is dependent on F2 in the prior load. When the load is finished executing and the result is available, then the tag on the multiply instruction indicates that the intermediate result from the load instruction is ready.); and if any of the input data values is invalid, setting the associated output data validity tag to indicate that the intermediate result is invalid (page 246 and 247, Rj and Rk are set to indicate the validity of a functional units input data that is output from the various functional units.).

***14.*** Referring to claim 15, Hennessy has taught a method in accordance with claim 14, as described above, and further comprising initializing an output data validity tag in the result register of each of the plurality of functional units to indicate that the associated intermediate result is invalid (page 246 and 247, Rj and Rk are set to NO.).

***15.*** Referring to claim 16, Hennessy has taught a method in accordance with claim 15, as described above, and further comprising: storing output data values only if the associated output data validity tag indicates that the data is valid (page 246 and 247, An operation is performed and the data is output only if the validity tags indicate that the data is valid.).

***16.*** Referring to claim 17, Hennessy has taught a method in accordance with claim 15, as described above, and wherein the processor further comprises at least one data source unit (page 246 and 247, FJ and Fk are the source values that come from source units.), each with an associated source iteration counter (page 240, "i" is a source iteration counter.), the method further comprising: initializing each source iteration counter (page 240, "i" is initialized to 1); and at each iteration of the specified number of iterations for which a data value is to be read by a data source unit: determining from the source iteration counter associated with the data source unit if all data values have been retrieved from memory (page 240, when "i" is less than or equal to 99, then all of the data values in the loop have not been retrieved from memory, when "i" is greater than 99, then all of the data values in the loop have been retrieved from memory); if not all data values have been retrieved from memory, adjusting the source iteration counter (page 240, "i" is adjusted each iteration.), retrieving a data value from a data memory

and setting the associated data validity tag to indicate that the result is valid (page 246 and 247, When an operation executes in a functional unit, dependent instructions mark the dependant operand as ready in respective the functional unit scoreboard.); and if all data values have been retrieved from memory, setting the output data validity tag to indicate that the result is invalid (Pages 246-247, Once an operand has been read the validity fields, Rj and Rk, are set to zero.).

***17.*** Referring to claim 18, Hennessy has taught a method in accordance with claim 17, as described above, and further comprising each data source unit signaling the controller when the associated source iteration counter indicates that all data values have been retrieved from memory (page 240, "i" is adjusted each iteration until the looping condition is no longer satisfied. When "i" reaches 100, then this indicates that all loop data values have been retrieved from memory.).

***18.*** Referring to claim 19, Hennessy has taught a method in accordance with claim 15, as described above, and wherein the pipelined program loop has no epilog instructions (This limitation is interpreted as "the loop body instructions exit a pipeline with instructions following the loop executing in the pipeline stages" (see the 112 rejection above). page 240, The loop exits the pipeline with at least one instruction following the loop in the pipeline, B[101] = C[100] + D[100];) and the processor further comprises at least one data sink (pages 246-247, Fi), each data sink being associated with a sink iteration counter (page 240, i) and operable to receive an output data value and an associated output data validity tag from the interconnection switch (page 246 and 247, Rj and Rk indicate the readiness of operands dependent on other operations

executed and output from the various functional units.), the method further comprising: initializing the sink iteration counter of each data sink (page 240, "i" is initialized to 1); and at each iteration of the specified number of iterations: determining the validity of the output data from the associated output data validity tag (page 240, page 246 and 247, Rj and Rk indicate the readiness of operands dependent on other operations executed and output from the various functional units. The validity of dependent instructions in a loop are checked each iteration.); if the output data is valid: determining from the sink iteration counter if all data values have been committed to memory (page 240, When a dependent loop instruction executes, the iteration counter is checked before the loop executes again.); adjusting the sink iteration counter associated with the data sink unit if not all data values have been committed to memory (page 240, I is incremented each cycle until all of the data values in the loop are committed to memory.); committing the output data value to memory if not all data values have been committed to memory (page 240, Data values are committed to memory during loop execution.); and signaling the controller if all data values have been committed to memory (page 240, the controller is signaled to stop executing the loop when I is greater than 99.).

***19.*** Referring to claim 20, Hennessy has taught a method for executing a pipelined program loop having a loop body but no epilog instructions on a processor (This limitation is interpreted as "the loop body instructions exit a pipeline with instructions following the loop executing in the pipeline stages" (see the 112 rejection above). page 240, The loop exits the pipeline with at least one instruction following the loop in the pipeline, B[101] = C[100] + D[100];) comprising a plurality of functional units (pages 246

and 247, at least the multiply unit, integer unit, add unit and divide unit are the functional units) and at least one data sink coupled through an interconnection switch and controlled by a controller (pages 246-247, Fi), each data sink being associated with a sink iteration counter (page 240, "i" is a source iteration counter.), the method comprising: initializing each sink iteration counter (page 240, "i" is initialized to 1.); executing the loop body for a specified number of iterations (page 240, executed I, or 99, times. ); and at each iteration of the specified number of iterations for which a data value is to be sunk by a data sink unit: determining if the sink iteration counter indicates that all data values have been committed to memory (page 240, "i" is checked each iteration to determine if all of the iterations have been executed, or if all data values in the loop have been committed to memory.).

***20.*** Referring to claim 21, Hennessy has taught a method in accordance with claim 20, as described above, and further comprising committing the data value to memory if not all data values have been committed to memory (page 240, when "i" is 1 to 99, then all instructions in the loop are executed and their data values are committed to memory.).

***21.*** Referring to claim 22, Hennessy has taught a method in accordance with claim 20, as described above, and further comprising adjusting the sink iteration counter associated with the data sink unit if not all data values have been committed to memory (page 240, "i" is incremented until the end of the loop, which is when all data values have been committed to memory.).

***22.*** Referring to claim 23, Hennessy has taught a method as in claim 20, as described above, and further comprising each data sink unit signaling the controller when the associated sink iteration counter indicates that all data values have been committed to memory (page 240, When "i" is incremented to 100, then all data values have been committed to memory and the controller proceeds to execute the instructions following the loop.).

***23.*** Referring to claim 24, Hennessy has taught a method as in claim 23, as described above, and further comprising terminating the execution of the pipelined program loop after all data sink units have signaled to the controller that all their data values have been committed to memory (page 240, When "i" is incremented to 100, then all data values have been committed to memory and the execution of the loop is terminated. The controller then proceeds to execute the instructions following the loop.).

***Conclusion***

24. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tonia L. Meonske whose telephone number is (571) 272-4170. The examiner can normally be reached on Monday-Friday with first Friday's off.

25. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Donald Sparks can be reached on (571) 272-4201. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

26. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

TLM

Tonia L. Meonske
May 31, 2007